Support Vector Machine

BA810: Supervised Machine Learning

Nachiketa Sahoo

Recap

ML Project Outline

1. Load and explore, training-test split
2. Consider cleaning and transformations
3. Create pipelines
4. Evaluate predictive models
5. Finetune hyperparameters of the most promising model
6. Estimate error on unused test-data

- Feature Selection
  - To improve prediction by reducing overfitting
  - Smaller models → easier to interpret

Feature Selection

Approach	Pros	Cons
Missing values and **variance-based**	Very fast, quickly eliminate many	No evaluation of prediction ability
Univariate: based on each feature's dependence on target	Very fast, almost always feasible	Pairwise dependence does not capture prediction ability in the presence of other features
Model based importance: keep important features according to a fitted model	Fast, scales to many features	Doesn't consider that removing one feature may change the effectiveness of others.
Recursive Feature Elimination: remove least important features one at a time, after refitting	Effect of prior feature removal on importance of each remaining feature is considered.	Importance may not correspond to the contribution towards prediction performance. Requires one fitting per removal (K times number of features, for K-fold CV to determine number of features to remove)
Stepwise search: Add/remove features based on contribution to prediction performance measured using cross validation	The *most accurate feasible approaches.* Can consider removal (addition) after each addition (removal) to further improve	Can be slow: may be infeasible with > 20–30 features. Forward search can miss complementary features. Backward search infeasible for linear regression when number of features > # of records.
Best subset selection: after evaluating all feature subsets	(Theoretically) finds the best features given unlimited data	Slowest: time increases exponentially with features. Overfits (in practice) given finite data. *Rarely used.*

Outline

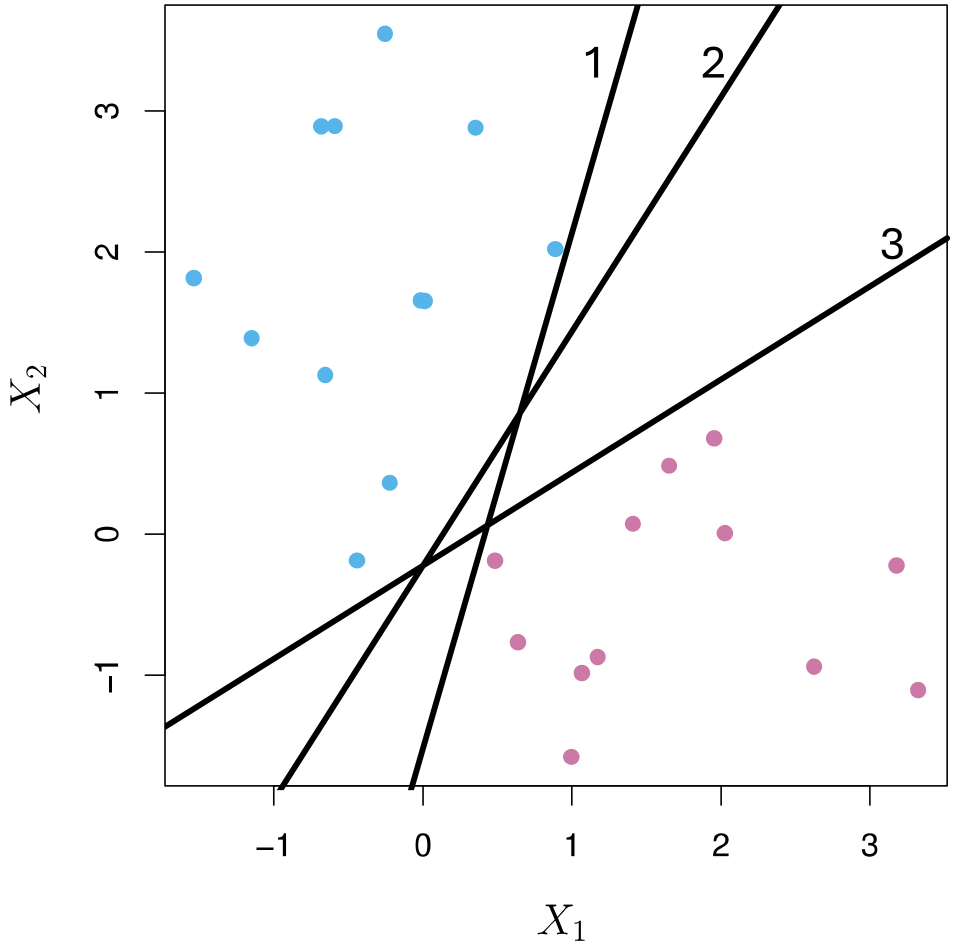
- Mostly linear models considered so far
  - Linear regression, Logistic regression
- Regularization to reduce overfitting, simplify (fitted curve/class boundaries)
  - Improve prediction by incurring a little bias to hopefully reduce variance a lot
- Feature selection is another way

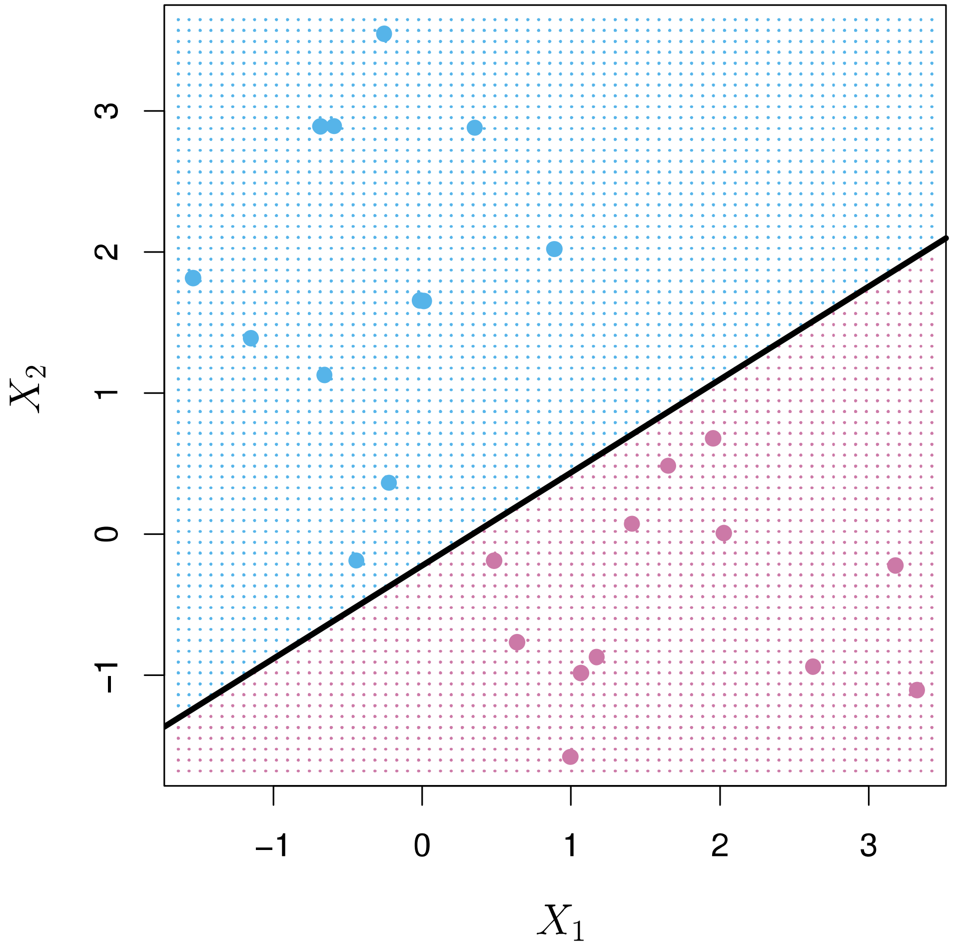
- What if the dataset is such that boundaries between classes aren't simple?
  - We *add* features (in a smart way)

Key ideas of SVM

1. It's good to have class boundaries that have large gap/margin around them
2. Data points more easily separate in higher dimensions (Kernel trick)

Maximum Margin Classifiers

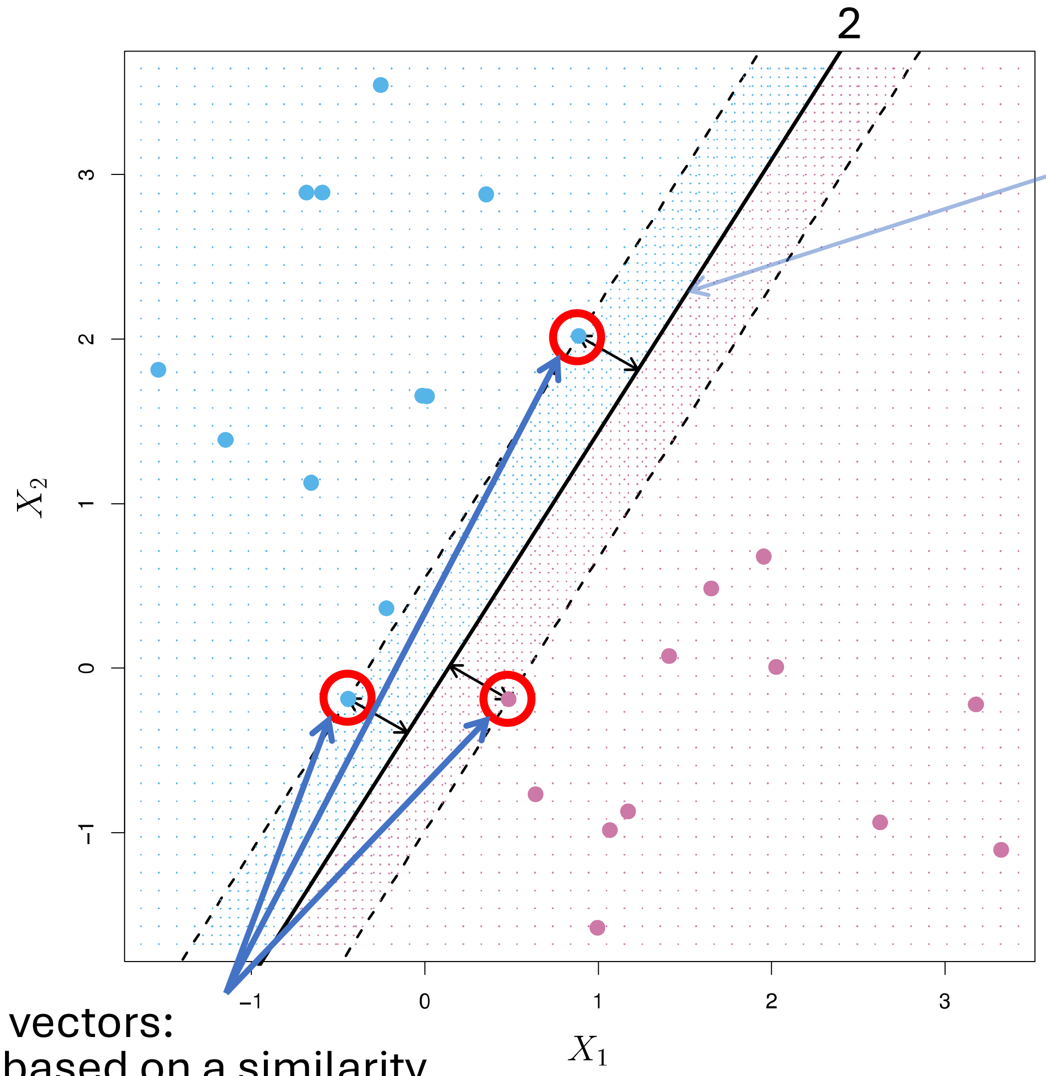




Effect of adopting the line 3 as the classifier

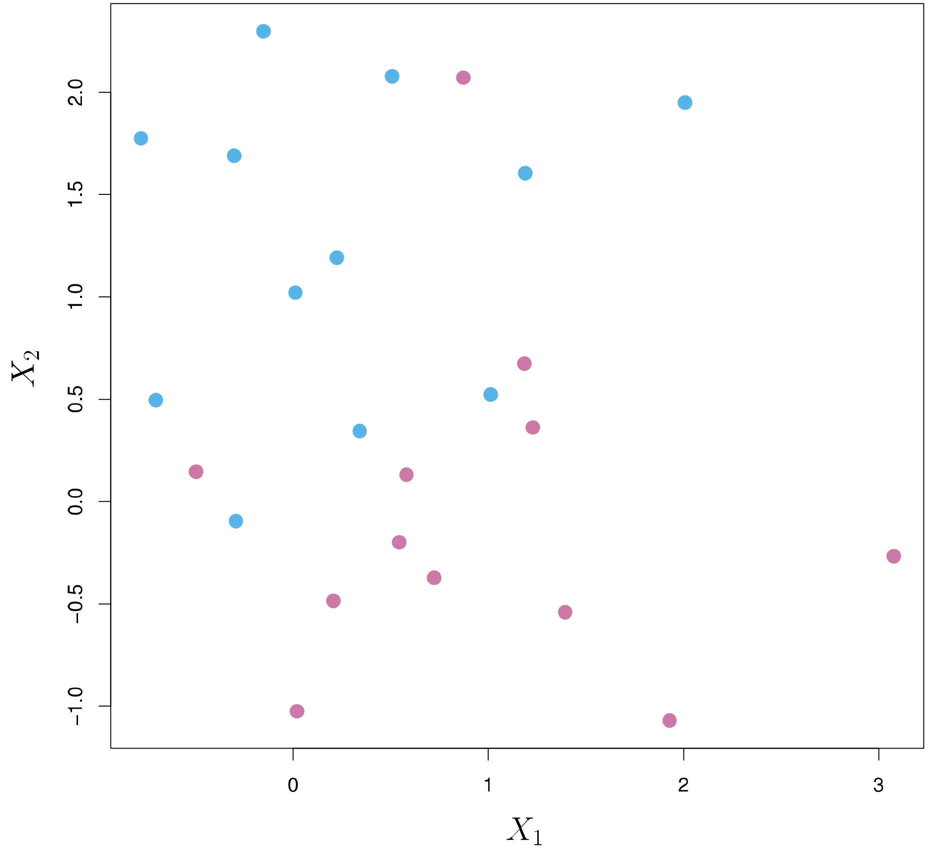
Which of 1, 2, and 3 is a better classifier?

Maximum Margin Classifiers

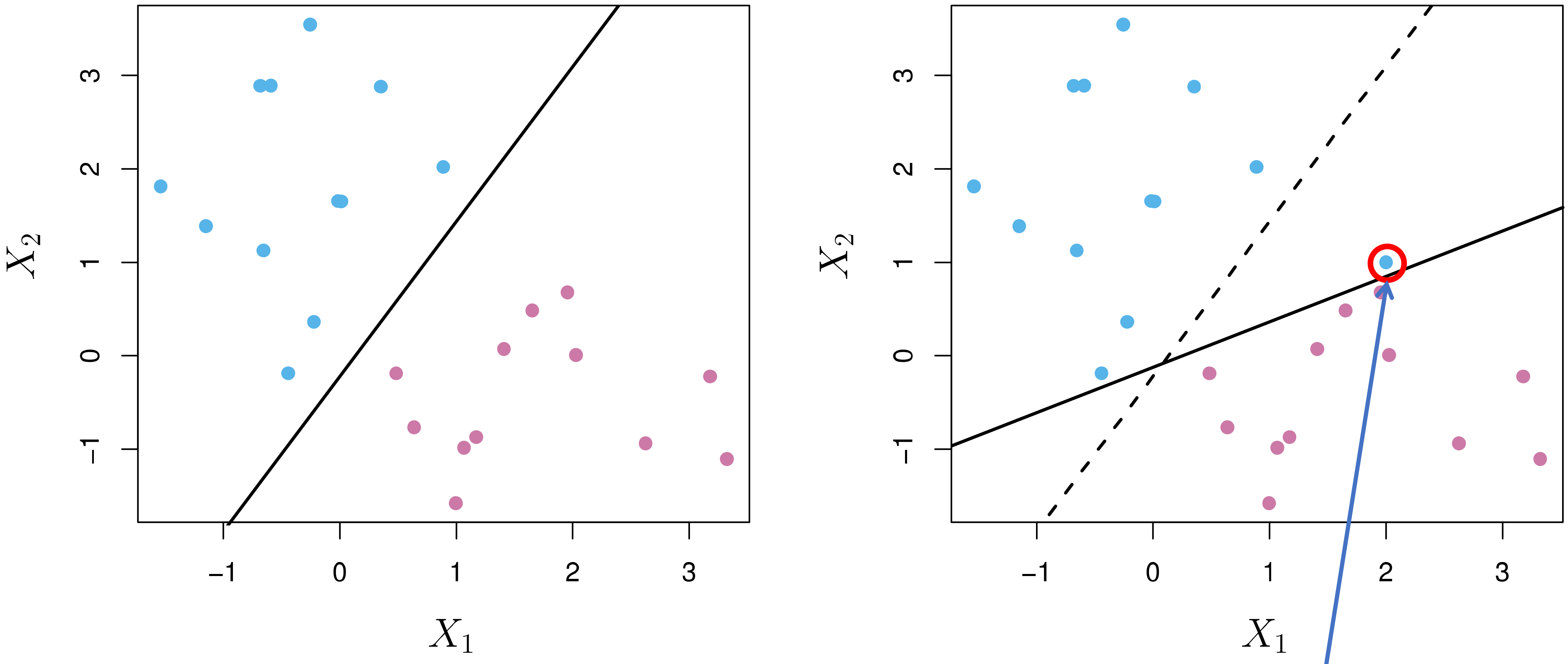


Support vectors:
classify based on a similarity
weighted combination of only these.

- Large margin during training → less chance of test data falling into the "wrong" side of the boundary.
- But what if you can't fit a line through the red and blue regions?



Or Data Are Noisy?

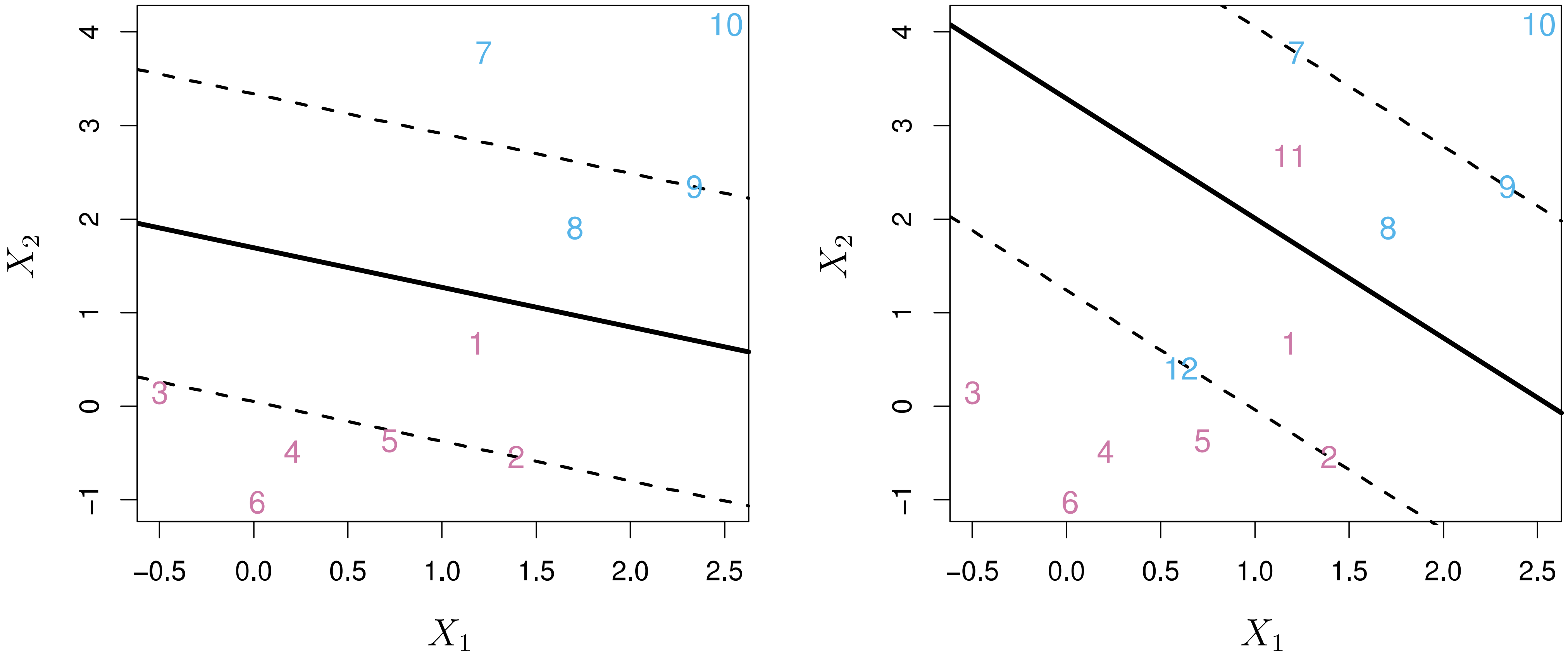


Typical set

Noise added

Use a Soft Margin

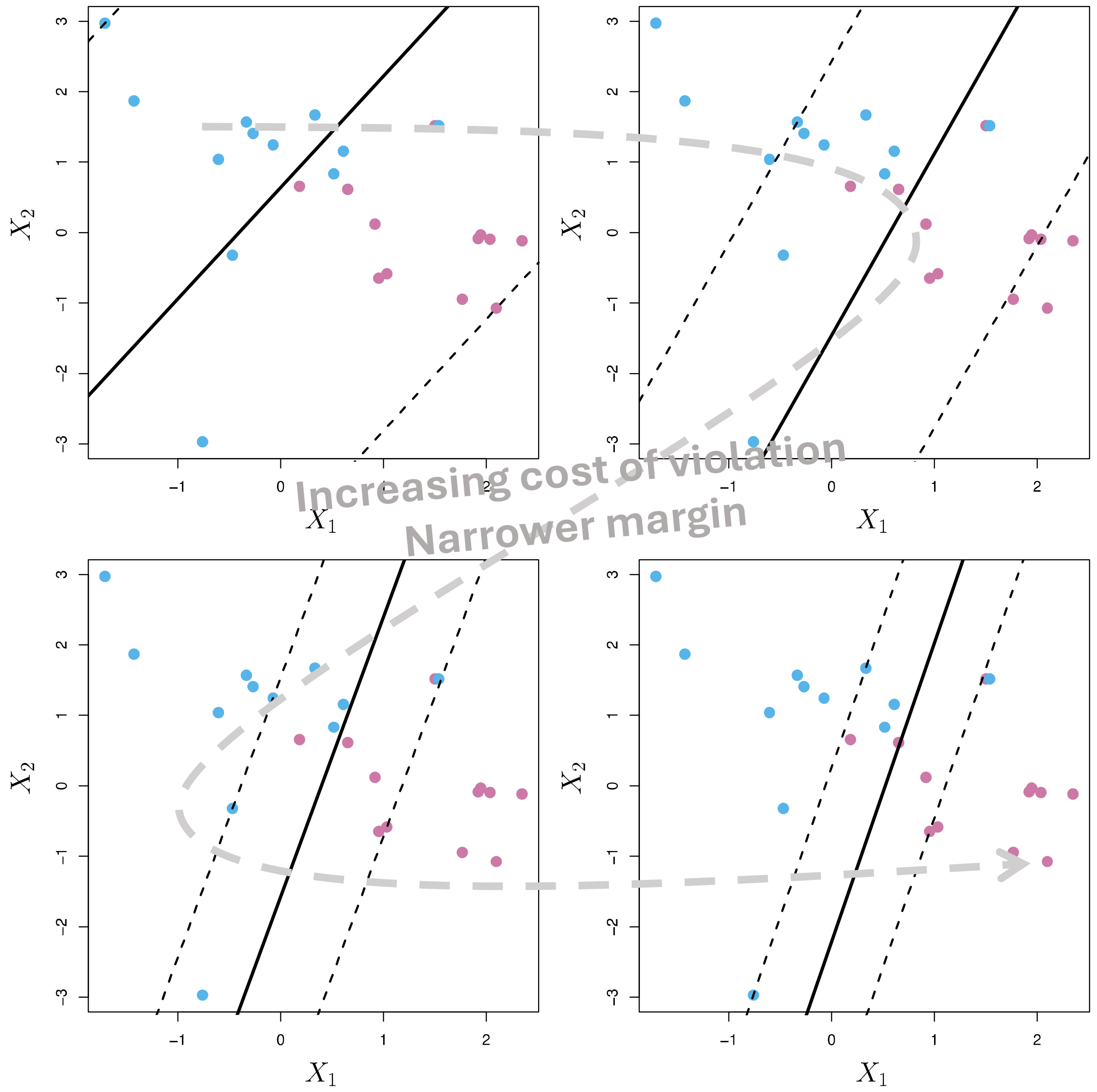
Allow Margin Violation at a Cost During Training



Per violation Cost specified by the parameter C in scikit learn (minimizes total cost).

What happens to the margin as we increase C?

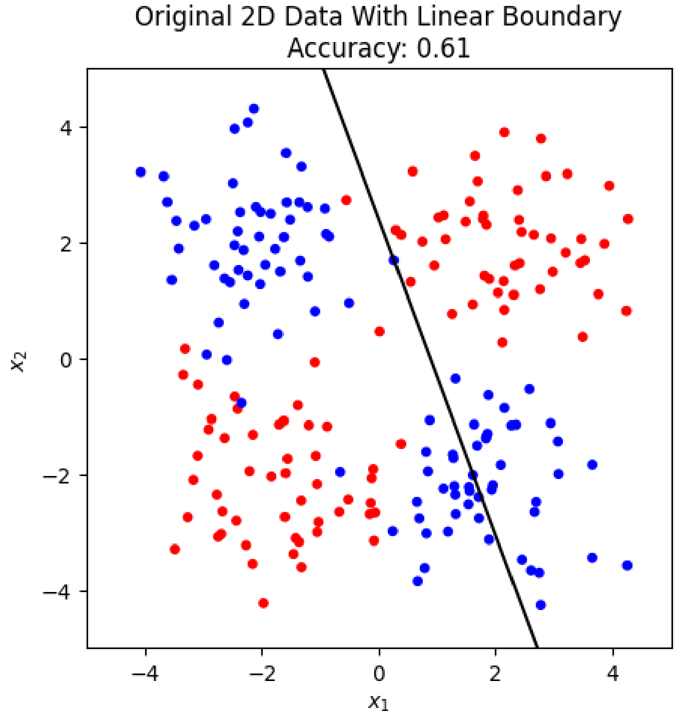
Effect of Cost of Violation on Margin



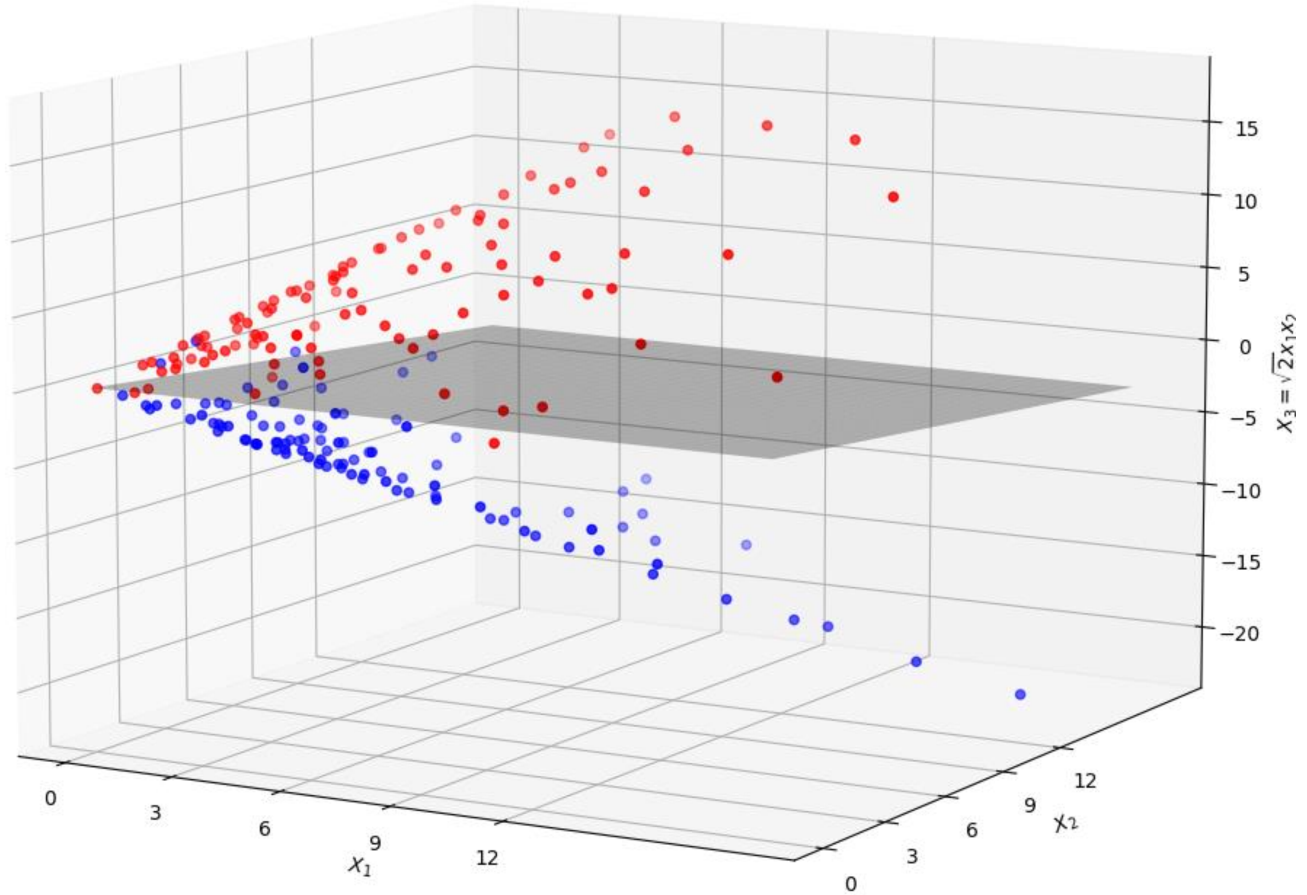
Increasing the cost of violation C tends to shrink the margin (to put fewer and fewer points within the margin).

- ∞ cost, no violation → original maximum margin classifier
  - Can fail to fit or overfit
- Reducing cost → regularization
  - Less sensitive to noise
  - Can underfit at the extreme
- Chosen using cross validation

But Sometimes a Linear Boundary is Just Unsuitable



Linear boundary in higher dimension, now a hyperplane, has accuracy of 0.95.

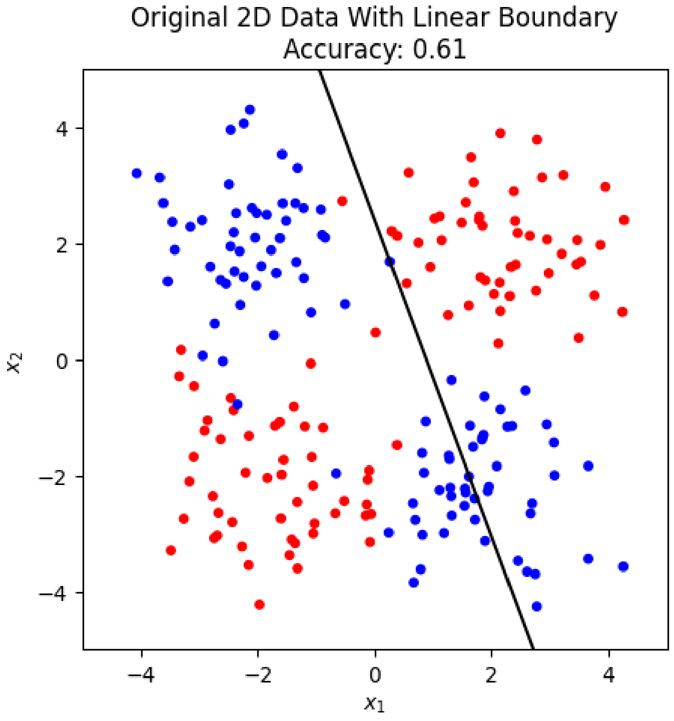


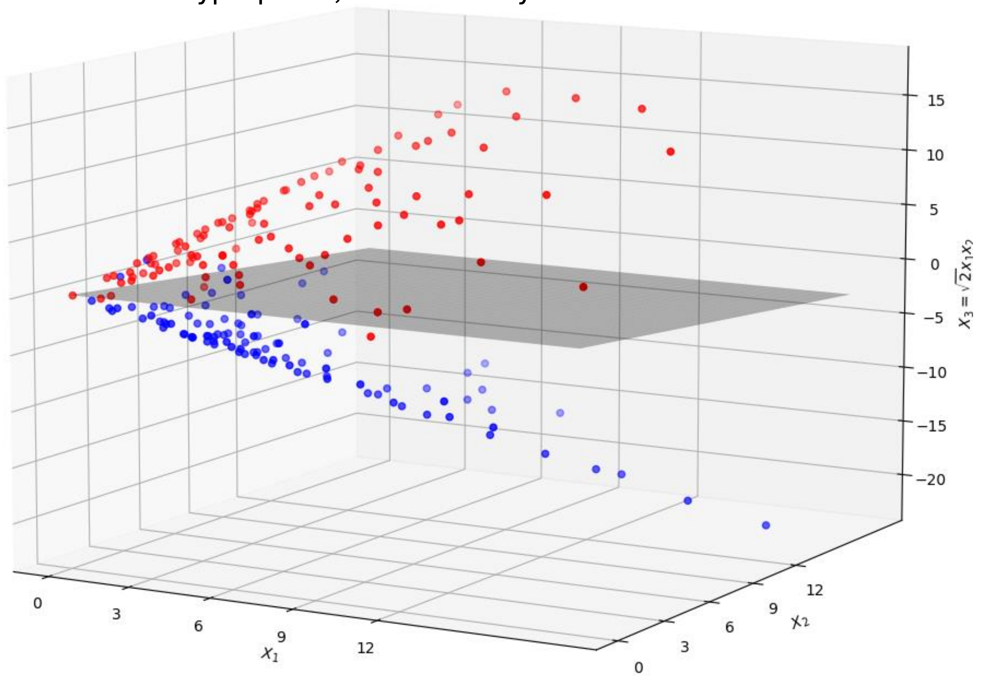
Create a dataset of three variables:

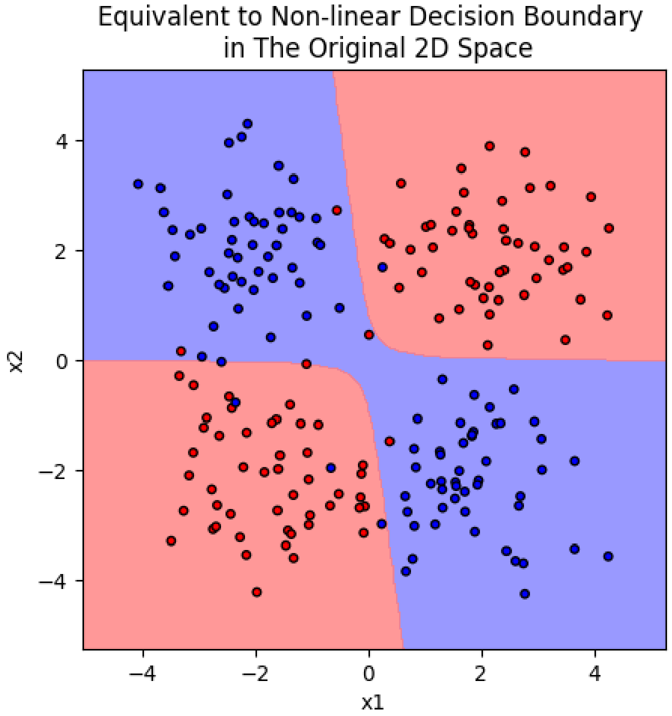
$$X_1 = x_1^2$$
$$X_2 = x_2^2$$
$$X_3 = \sqrt{2}x_1x_2$$

But Sometimes a Linear Boundary is Just Unsuitable

Linear boundary in higher dimension, now a hyperplane, has accuracy of 0.95.

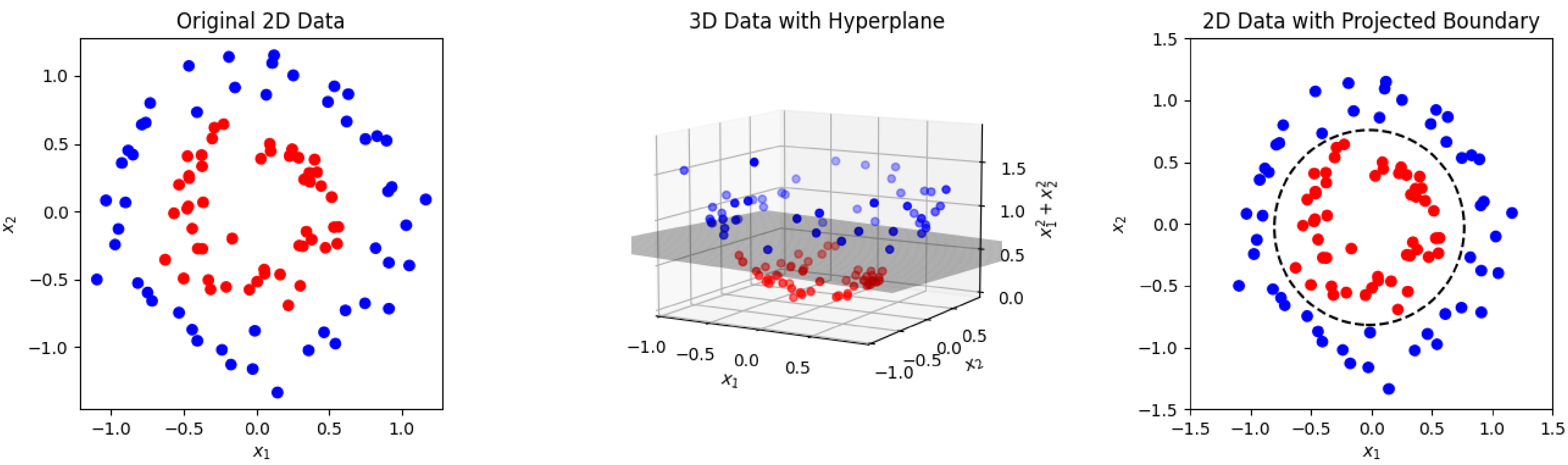






The points at the boundary have the same values of X_3.

Another Simple Example



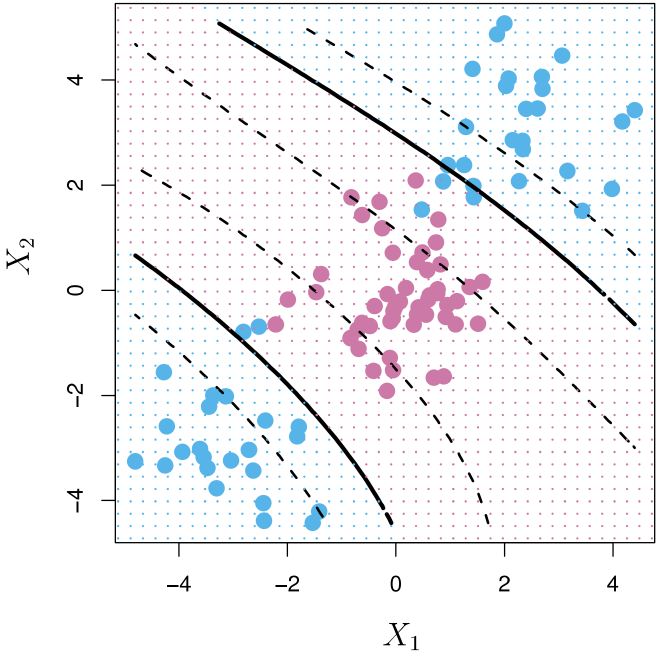
- This projection was carefully chosen
- We'll often let the algorithms do that for us, implicitly through "Kernel" functions.

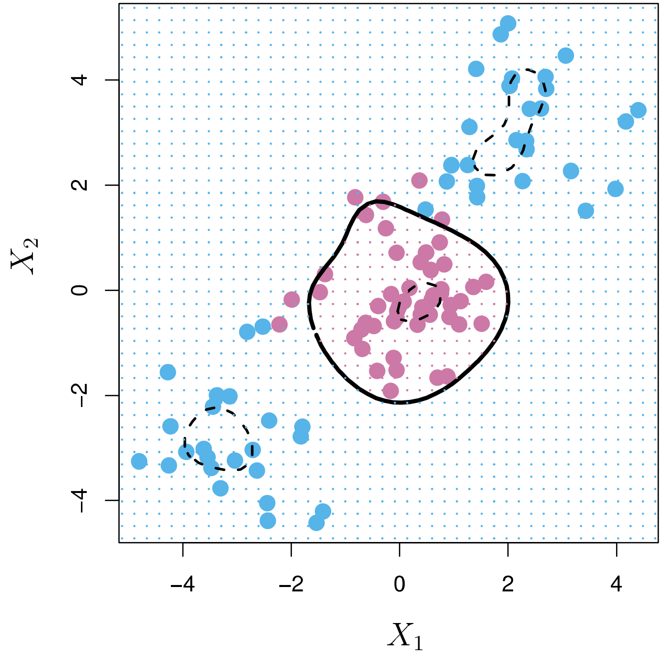
The points at the boundary have the same values of $X_3 = x_1^2 + x_2^2$, thus a circle.

Kernels

- Recall: need only similarities to support vectors to classify
- Kernels $K(\boldsymbol{x}, \boldsymbol{x_i})$ compute similarity that's equivalent to adding higher order terms and then taking dot product: $\boldsymbol{x} \cdot \boldsymbol{y} = (x_1, \dots, x_p) \cdot (y_1, \dots, y_p) = \sum_{j=1}^{p} x_j y_j$
- Simplest is Linear Kernel: $K(\boldsymbol{x}, \boldsymbol{x_i}) = \boldsymbol{x} \cdot \boldsymbol{x_i}$
- Polynomial kernel: $K(\boldsymbol{x}, \boldsymbol{x_i}) = \left(1 + \sum_j^p x_j x_{ij}\right)^d$
- Radial (basis function, RBF) kernel: $K(\boldsymbol{x}, \boldsymbol{x_i}) = \exp\left(-\gamma \sum_j^p \left(x_j - x_{ij}\right)^2\right)$
  - Equivalent to ∞ dimensional representation
  - Large $\gamma \geq 0$ → similarity falls away with distance faster

Polynomial kernel of degree 3



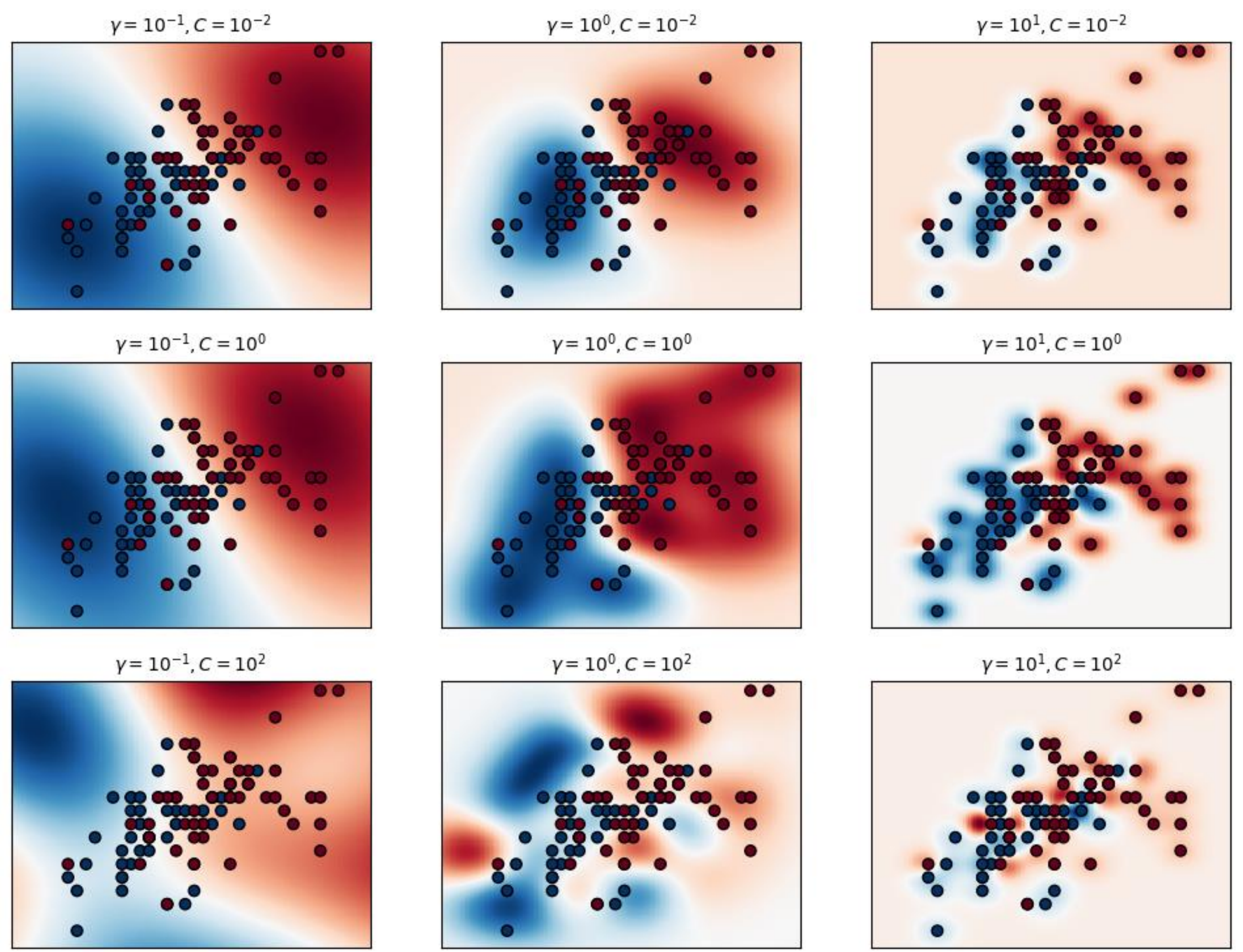


Radial kernel

Using RBF Kernel
Role of C and γ

The color shows the classification (blue or red) of possible points, and the intensity is confidence in that classification.

1. As C increases (down a column), class boundary becomes more complex.
2. As γ increases (right on a row), the region that gets blue or red classification shrink to near the training points.
3. Optimum value of C might depend on that of γ (and vice-versa).
    1. Important to set them jointly using hyper-parameter search.
    2. Best C for RBF may not be best for linear or polynomial kernel.



General Tips

- For datasets with large number of features use linear kernel
  - Much faster and additional benefit from higher order terms is small
  - No general guidelines, but you may find with >15 features linear is better than RBF
- For fewer features and relatively small datasets
  - Radial kernel (often works better than polynomial kernel)
  - Need to search for C and γ parameters (for radial kernel) jointly
- Start with linear kernel, then consider RBF

- Additional resources
  - (Easier) Datacamp tutorial article on SVM using scikit learn
  - (More challenging) Andrew Ng's course video on SVM
  - (More challenging) Kernel chapter from Joaquin Vanschoren's book